

Working with Configuration Providers



Steve Gordon

MICROSOFT DEVELOPER TECHNOLOGIES MVP

@stevejgordon www.stevejgordon.co.uk



Overview



Configuration providers

How configuration is populated

Configuring applications using environment variables

Configuring applications using command line arguments

Securing configuration secrets

- Development - User secrets
- Production - Azure Key Vault

Configuration from AWS Parameter Store

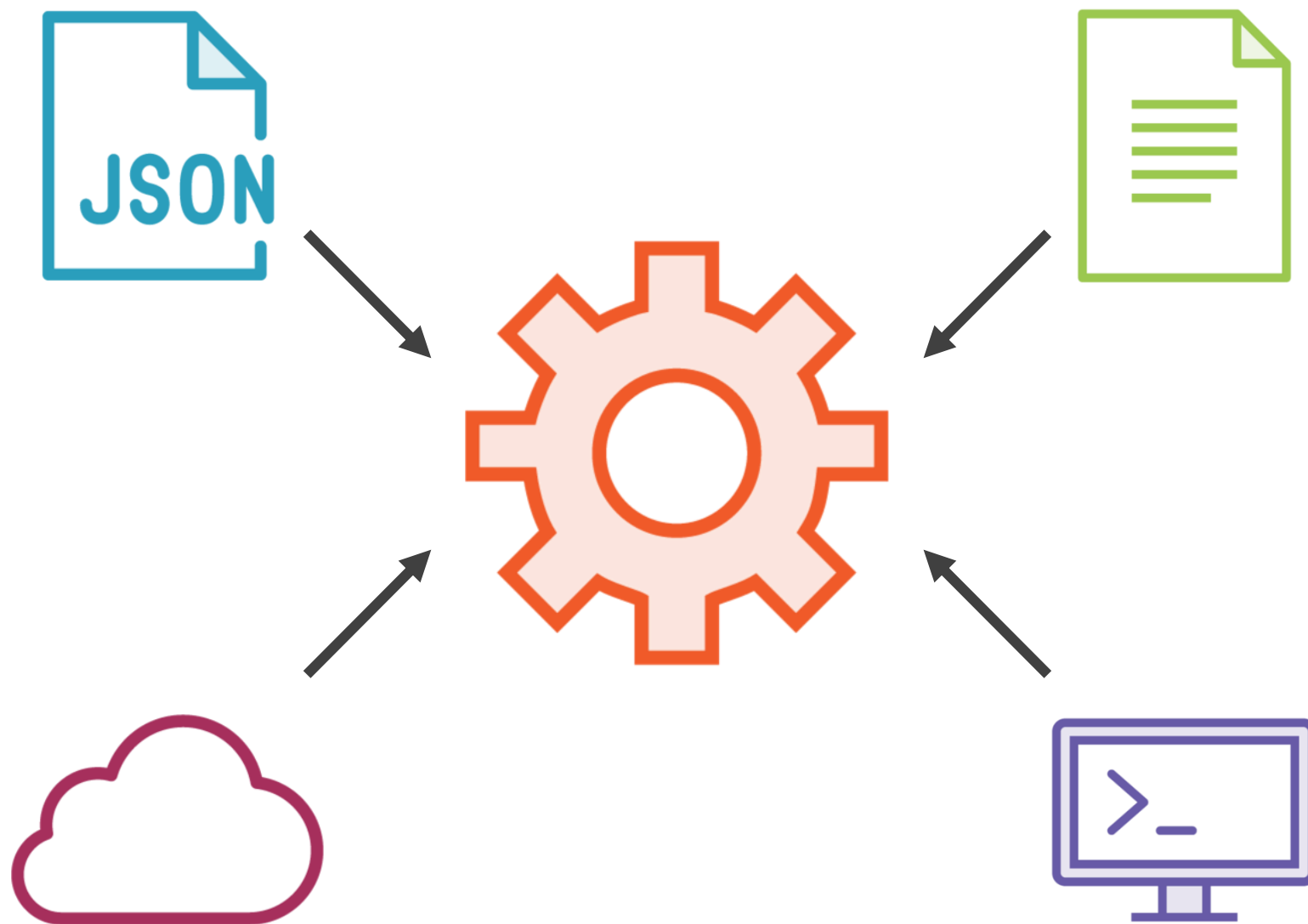
Controlling configuration provider order

Defining a custom configuration provider

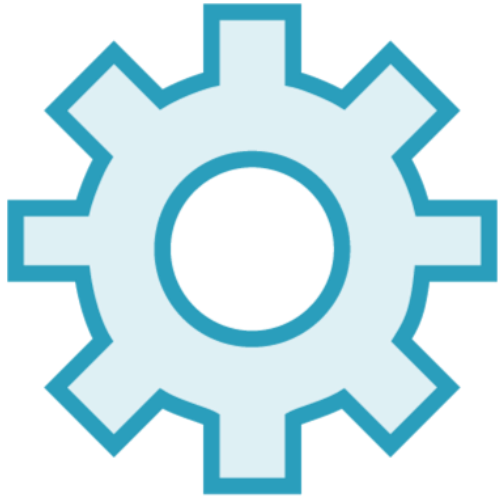


Configuration Providers

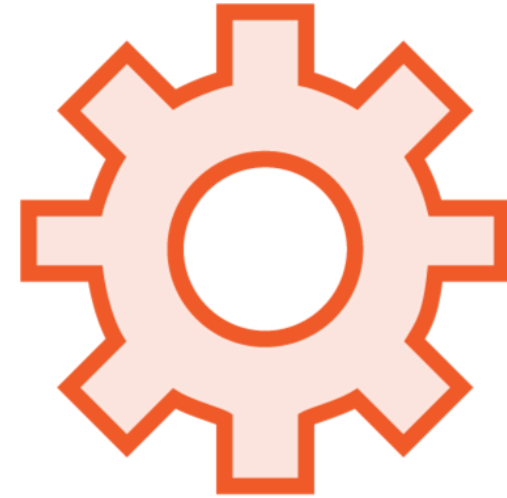




Where Configuration Applies



Host configuration



Application configuration

Configuration Providers

Executed in order to load configuration

Key/values pairs are added or replaced by each configuration provider

The order of the configuration providers affects the final configuration values

A set of default providers are added by the host builder



public static IHostBuilder CreateDefaultBuilder(string[] args)

```
builder.ConfigureAppConfiguration((hostingContext, config) =>
{
    var env = hostingContext.HostingEnvironment;

    config.AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
        .AddJsonFile($"appsettings.{env.EnvironmentName}.json", optional: true, reloadOnChange: true);

    if (env.IsDevelopment() && !string.IsNullOrEmpty(env.ApplicationName))
    {
        var appAssembly = Assembly.Load(new AssemblyName(env.ApplicationName));
        if (appAssembly != null)
        {
            config.AddUserSecrets(appAssembly, optional: true);
        }
    }
    config.AddEnvironmentVariables();
    if (args != null)
    {
        config.AddCommandLine(args);
    }
})
```



How Configuration Is Populated



appSettings.json

```
{  
  "HomePage": {  
    "Title": "A Good Title",  
    "ShowGallery": true  
  },  
  "MaxNewsItems": 10  
}
```

→ **HomePage:Title = "A Good Title"**

→ **HomePage:ShowGallery = true**

→ **MaxNewsItems = 10**



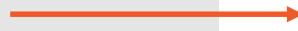
appSettings.Production.json

```
{  
  "HomePage": {  
    "Title": "A Better Title" → HomePage:Title = "A Better Title"  
    → HomePage:ShowGallery = true  
  },  
  "MaxNewsItems": 10, → MaxNewsItems = 10  
  "HideAds": true → HideAds = true  
}
```



Environment Variables

HomePage__ShowGallery = true



HomePage>Title = "A Better Title"

HomePage:ShowGallery = true

MaxNewsItems = 10

HideAds = false



HideAds = false



Demo



Providing configuration from environment variables

- Defining configuration using environment variables
- Defining environment variables using `launchSettings.json`



Demo



Passing configuration using
command line arguments



Securing Sensitive Data in Configuration



```
{  
  "HomePage": {  
    "Title": "A Good Title",  
    "ShowGallery": true  
  },  
  "MaxNewsItems": 10,  
  "UsersApiKey" : "SUPERSECRET"  
}
```

◀ This title is not sensitive

◀ This poses a security risk



Developer Responsibility



Security is always a priority

Consider the sensitivity of configuration values if exposed

Avoid exposing sensitive data within insecure sources

Code reviews should highlight security risks

A true story!



Keep your secrets, secret!



Demo



Securing secrets in development

- User secrets CLI
- Managing user secrets



Demo



Securing secrets in production

- Azure Key Vault



Demo



Loading configuration from AWS

- Defining configuration using Systems Manager: Parameter Store
- Adding the AWS Parameter Store configuration provider



Demo



Customizing the order of configuration providers

- Clearing the default providers
- Registering providers using the `IOptionsBuilder`



Demo



Load configuration from a database

Define a configuration provider

Define a configuration source

Register the custom provider with
the IConfigurationBuilder



Review



Default configuration providers

Configuring an application using:

- Environment variables
- Command line arguments

Secure configuration using:

- User Secrets
- Azure Key Vault

**Loading configuration from AWS
Parameter Store**

**Customizing the order of
configuration providers**

Defining a custom configuration provider

